Provisioning of Grid Middleware for EGI in the framework of EGI-InSPIRE

M. David[1], G. Borges[1] , J. Gomes[1], I. Campos[2], A. Lopez[2], P. Orviz[2]
J. López Cacheiro[3], C. Fernandez[3] and A. Simon[3]

[1] Laboratório de Instrumentação e Física Experimental de Partículas Lisboa, Portugal
david@lip.pt
[2] Instituto de Fisica de Cantabria, CSIC, Santander, Spain
[3] Fundacion Centro de Supercomputacion de Galicia, Santiago de Compostela, Spain

Abstract. The *European Grid Initiative* (EGI) will provide a sustainable pan-European Grid computing infrastructure for e-Science based on a network of regional and national Grids. Although EGI will emerge from the structure established by the project *Enabling Grids for E-SciencE* (EGEE), the organizational and operational aspects will be deeply changed. The transition to the new model will also impact on the middleware development and deployment issues. Under the new model, Ibergrid is now coordinating the middleware deployment processes in EGI. This paper describes the EGI middleware strategy and the middleware rollout workflow being planned by Ibergrid for the whole European infrastructure.

1 Introduction

The *European Grid Initiative* (EGI.eu) [1, 3] aims to setup an organization that will enable the sustainable provision of grid computing services to the research community in Europe. EGI will provide a mechanism whereby national resources will be fully integrated into a pan-European infrastructure. The mechanism and interfaces for resource integration and software provision for users and application support will form the basis of the EGI coordination work. To this end EGI will provide the necessary middleware by partnering with several external software providers.

In the current picture it is foreseen that external software providers will release middleware and software technology potentially interesting for EGI. In certain cases the software will have been developed at the request of EGI, motivated by user requirements or operational requirements.

To manage the software adoption process, EGI introduces the concept of Unified Middleware Distribution (UMD). The UMD defines a set of functional specifications, performance and quality requirements that the software registered in the EGI repositories must fulfil.

EGI will select from these repositories an integrated distribution of middleware and software services and release to the infrastructure providers for installation, this defines a particular UMD release. A Roadmap for the UMD will be devised in

order to provide an mechanism for service upgrading, improvement, and inclusion of new functionalities.

The external software providers will be identified and an agreement between EGI and the provider will establish the formal relationship between the software provider and EGI. This will be described in a Operational Level Agreement (OLA) which will include the agreed release schedule and expected support and maintenance of the software components.

The software providers will be either UMD software providers, EGI Software providers and Community contributors. The first ones will have agreed to provide EGI with a set of components and specified release dates to fulfil the UMD roadmap requests.The project European Middleware Initiative (EMI) and the Initiative for Globus in Europe (IGE) are obvious candidates to provide the software necessary for the UMD releases.

The software provided within EGI (by the National Grid Initiatives) which is not part of the UMD will also undergo a verification and validation mechanism prior to the rollout in the production infrastructure in EGI, although its adoption will be completely up to the NGIs. On the same level, community software projects providing software which might be of interest to EGI will find a place in the EGI software repositories, provided the verification and validation of their software has passed the EGI processes.

The FP7 project EGI-InSPIRE [2] will undertake the migration of the European Grid e-Infrastructure and its services to the new sustainable governance model. The software provision activity in EGI-InSPIRE (inside the work package SA2) and the rollout into production activity task (coded as TSA1.3) fall under the responsibility of NGI-ES and NGI-PT.

The overall EGI service life cycle is represented in Figure 1.

The global software provisioning task can be referred as service transition inside *Information Technology Infrastructure Library* (ITIL) terminology.

In this paper we will focus in the service transition stage describing the processes and functions involved.

The Grid Middleware Process in EGEE is briefly described in section 2. Section 3 describes the plan for the transition from EGEE to EGI-InSPIRE. In section 4 it is given an overview of the Middleware Unit tasks *Definition of the Unified Middleware Distribution quality criteria* and *Verification of conformance criteria*, while section 5 describes the Operation Unit task *Service Deployment Validation*. Conclusions are given in section 6.

2 Grid Middleware Rollout Process in EGEE

The EGEE project produces a Grid middleware stack called *gLite* [4], composed of several components. Some of these components are developed and supported externally to EGEE, while others are developed and supported in a dedicated activity inside the project.

Having many components from several development teams (not all under the EGEE coordination), leads to a rather complex process to produce a single middleware stack such as gLite. The integration process of all components requires a

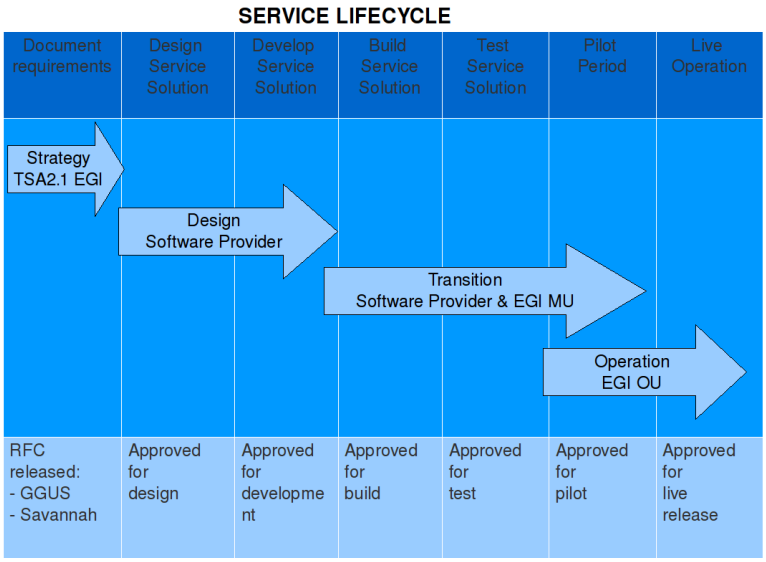

Fig. 1. EGI service lifecycle

stringent certification and verification procedure, in order to assure that all components are interoperable and interact in the correct way. There is an EGEE activity dedicated to this process.

Experience over the last years has shown that in many cases the gLite version deployed in production after integration and certification from integration team, contains errors or bugs impacting on one or more Grid components. The controlled environment on which the integration team preforms the testing integration and certification of a given gLite release, is far from the large number of users, applications and wokflows with more chaotic and heterogeneous environment found in the production infrastructure.

One can say that bugs were hidden to the homogeneous integration environment, but showed themselves in the heterogeneous production infrastructure.

To tackle these issues, the EGEE Operations Team, created an infrastructure independent from the production one, called *Pre-Production Service*. This infrastructure was deployed by production sites with a larger number of resources and more heterogeneous than the integration service.

The aim of *Pre-Production Service* was to expose new middleware versions to an environment closer to the production one. This early deployment was open to site administrators and users which could test the new middleware versions with their applications and workflows. It would allowing the eventual discovery of bugs or issues which would otherwise cause a negative impact on the production infrastructure.

Despite a rather big effort from the sites participating in the *Pre-Production Service*, which included:

– Deployment of new services.
– It's operation.
– Fast deployment of new middleware releases.
– Testing, monitoring and checking that no major issues or bugs are present.

The users and Virtual Organizations never made much usage of the *Pre-Production Service*, this was due to the fact that their production environment: software installation, data sets, workflows, were difficult and time consuming to reproduce or replicate in the *Pre-Production Service*.

The lack of usage of the *Pre-Production Service*, as been partially the cause of why some bugs have passed undetected to the production infrastructure, causing sometimes a disruption in the services. Nonetheless, one has to note that the *Pre-Production Service* has allowed the detection of many issues and bugs that would otherwise pass undetected into the production infrastructure. Furthermore, it is perceived that even if the *Pre-Production Service* would be more extensively used by production users, would not be a guaranty of a full proof version of the middleware components.

The low usage of the *Pre-Production Service* by production users, has lead the project to question the maintenance of such infrastructure, specifically the balance between the effort in human resources and the gain in robustness of the middleware releases.

On the other hand, experience shows that a buffer is needed for new middleware versions between certification and wide deployment in production.

As such, a different process to roll-out new versions of the middleware as been devised that on the one hand decrease the effort in resources, while allowing at the same time to improve the robustness and detection of bugs before wide deployment in the production infrastructure.

The two processes presently in place are called: *Staged Rollout* and *Pilot Service*.

2.1 Staged Rollout

The Staged Rollout is the operational process used to deploy middleware updates in a controlled way on the production infrastructure. The aim is to have a few sites that are the Early Adopters deploying new middleware versions, and give first hand feedback on it's experience.

Currently, the Staged-Rollout is performed in Pre-Production for some services, and in the production infrastructure for other services. This is due to the transition phase occurring at the moment, while sites decommission their services in the Pre-Production and deployed them in the production infrastructure.

New versions of the middleware components should thus be exposed in the production environment and if bugs or issues are found in this phase they can be mitigated and avoid widespread deployment in the infrastructure. The few sites that deployed the new problematic versions can either:

– Rollback the service to the previous working version, making a recommendation that the new version be rejected for wide deployment.

– The site can work in close collaboration with the integration team and/or developers in order to solve the problem. If the site has other instances in production which are working correctly, than it can be of the interest of the site and/or users communities to maintain a problematic version of the service in order to work more effectively on the problem.

The Grid production services running within this program are conventionally referred to as *beta services*, in the sense that they are object to special observation during the rollout and to special filtering in terms of site availability and reliability metrics.

2.2 Pilot Service

The *Pilot Service* is a process through which Grid services are set up on demand in order to let users communities and/or Virtual Organizations to interact with a new or changed middleware product or feature in a production context. They have a finite duration in time and measurable objectives (success criteria).

This process is particularly useful when, for example, there is a new service which will need strengthening and improvement of robustness, in environments requiring very high loads for that service. In such cases, some Virtual Organizations with specially demanding requirements should participate in those tests, closely working with sites and developers in order to improve the middleware component.

3 Transition process from EGEE to EGI-InSPIRE

The transition to a sustainable Grid infrastructure is being worked out by EGI.eu [2] in collaboration with the EGEE project. From the point of view of EGEE the project will finish on the 30 of April of 2010 and the EGI-InSPIRE project will start on the 1^{st} May 2010. From the operational point of view, no cut off or disruption should be felt either by sites or users of the infrastructure. A smooth transition and careful planing is needed for most of the tasks when changing from EGEE to EGI-InSPIRE.

One of the main differences of EGI-InSPIRE with respect to EGEE, is that the middleware will be produced and maintained externally to the project.

Figure 2 shows the workflow of the middleware from it's development phase until wide deployment in the production infrastructure.

The top box shows the "*Software Provider*", which as already stated in the introduction, can be an external project, a community of software developers, or can refer to a NGI developing software for its own needs.

Whatever the case, any middleware component will have to undergo the middleware testing and verification workflow depicted in Figure 2.

Particularly interesting is the case of the providers of software necessary to build the UMD, i.e. the official EGI release. The FP7 project European Middleware Initiative (EMI) will be providing the necessary services for the glite, ARC and Unicore software stacks.

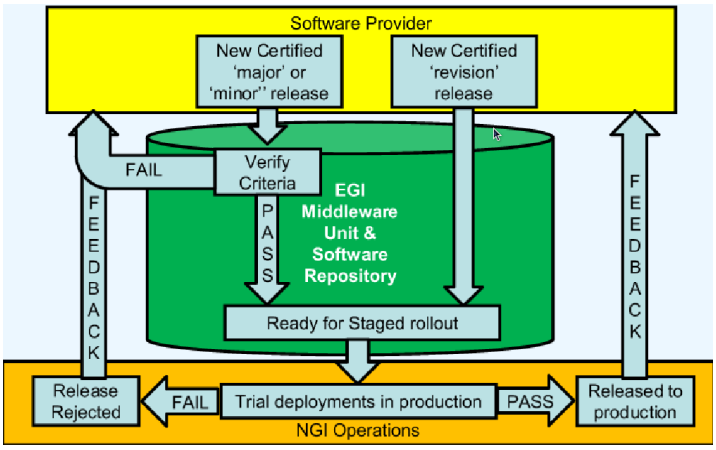

Fig. 2. Workflow of Middleware in the EGI-InSPIRE framework

This paper focus only in the gLite middleware. For the other middleware stacks, ARC and Unicore, similar procedures will have to be devised. Inside EMI each service is developed by a so called *Product Team* [5]. For example there will be product team providing the services necessary for Virtual Organizations Management Services, another product team will take care of MPI support and so on.

The middleware workflow from development to production in the EGI-InSPIRE is significantly different from the EGEE workflow. While in EGEE the certification and integration of all middleware components are performed by specific activities of the project, in the new era, the certification is performed by each *Product Team*, and the integration of the components is performed by a dedicated team in EMI.

The interaction between EGI-InSPIRE and EMI is done through the *Middleware Unit* activities of the EGI-InSPIRE project, described in section 4 below. The transition phase is currently underway, and should be mostly completed before the end of the EGEE project, so that when time comes, the new workflow is in place, or at least in an advanced state.

Another important piece of the transition puzzle is how the sites participating in the Pre-Production/Staged Rollout, will plan the transition. A survey is being conducted through the sites and NGI's, in order to assess the commitment to this task in the EGI-InSPIRE era.

As previously mentioned, the decommissioning of all sites and services in the *Pre-Production Service* infrastructure is presently underway, and the aim is to perform the Staged Rollout of all services in the production environment.

There are several situations that can occur for the sites currently participating in the Staged Rollout process:

– The site will be decommissioned near the end of EGEE III project, with no foreseen continuation in the EGI era.

– The *Pre-Production Service* will be decommissioned. The site commits with the same services or others to the Staged Rollout process. It will keep the commitment in EGI-InSPIRE as it has done in EGEE.
– In EGI, the focus are the NGIs, as such any given NGI can organize with their sites the level of commitment and participate in this task.

4 Middleware Unit in EGI-InSPIRE

Among the *Middleware Unit* tasks, there are two which are coordinated by Ibergrid, namely:

– TSA2.2: Definition of the *Unified Middleware Distribution* (UMD) quality criteria.
– TSA2.3: Verification of conformance criteria.

4.1 Definition of the UMD quality criteria

The definition of the UMD quality criteria is an activity that requires tight collaboration between EGI and the Software Providers in order to formalise the requirements for middleware components. Namely in providing and accepting specific tests and use cases for each component. Automated tests should be developed and implemented as part of the process, though manual tests will always remain.

The requirements will be specified as either generic acceptance criteria which should hold for any component in UMD, such as; interoperability, extensibility, availability on a specified minimal set of platforms, availability of SDK, security, requirements on documentation.

There may exist specific criteria applicable only for some middleware components, such as; throughput or behaviour under high load.

It is expected that the criteria for the acceptance of generic components will become stricter over time.

4.2 Verification of conformance criteria

The verification process depends on the type of release. Common traits are a report containing the results of the automated and non-automated tests defined in the task TSA2.2 (previous section) and documentation. The level of the tests and documentation will depend on the type of release:

– Major release or new component: introduces significant functionality changes having a greater risk of undiscovered defects. Verification of new functionality or interface changes against established criteria. The verification process is based around manual testing and the development of automated test suites involving operations, the users community and the software provider, wherever possible.
– Minor release: The availability of test suites and a test report will allow a streamlined verification process though some manual testing may still be perform.

– Revision and Emergency releases: Self-certification by the software provider that all bugs have been fixed and functional interfaces and behaviours remain unchanged. The quality assurance process from the software provider is relied upon.

5 Operations Unit in EGI-InSPIRE

This section details the plan on how to perform the "*Middleware Rollout*" to the production infrastructure, in the EGI-InSPIRE era. The task will be carried out by the Operations Unit task TSA1.3 and coordinated by Ibergrid.

Figure 3 shows the workflow of new middleware versions. The Software Providers push new middleware versions to the EGI-InSPIRE repository called *Scratch*, notifying the EGI-InSPIRE Middleware Unit. The EGI-InSPIRE Middleware Unit performs the process detailed in section 4 and in case of success, push the middleware components from the *Scratch* to a *Beta* repository.

The Operation Unit takes over in this phase, starting the Staged Rollout process, and at the end will push the new middleware versions to the production repository for wide deployment.

5.1 Versioning scheme and middleware components update

A new version of any middleware component falls on one of the following categories:

1. *Emergency*: bug fixes, or security vulnerability, backward compatible.
2. *Revision release*: bug fix, backward compatible.
3. *Minor release*: new functionality, backward compatible.
4. *Major release*:
   (a) New functionality not necessarily backward compatible.
   (b) New service.

Any middleware component can be updated only up to a *Minor Release* version while keeping the same EMI *Major Release* version. A "*Major Release*" of any component may only be included in the next EMI *Major Release*.

All categories of updates to the components will undergo the Staged Rollout process, though the timelines and depth of the process may vary with the category. The sole exception is an *Emergency* release, for which, under exceptional circumstances to be evaluated in a case by case basis, may skip the *Staged Roll-out*.

EGI-InSPIRE will accept only certified and validated updates provided by the *Product Teams*. The validated components will undergo the Staged Rollout process, and if successful can then be widely deployed in the production infrastructure.

In the Staged Rollout phase, if bugs or issues are found in a given component for which some solution or workaround is proposed, the fix or fixes should be communicated and implemented by the *Product Teams*. Middleware components with workarounds to bugs or issues should be avoided in production.

It is the aim of EMI to release middleware components centred on services, which may affect one or more node types, as opposed to the situation in EGEE where the releases are centred on node types.

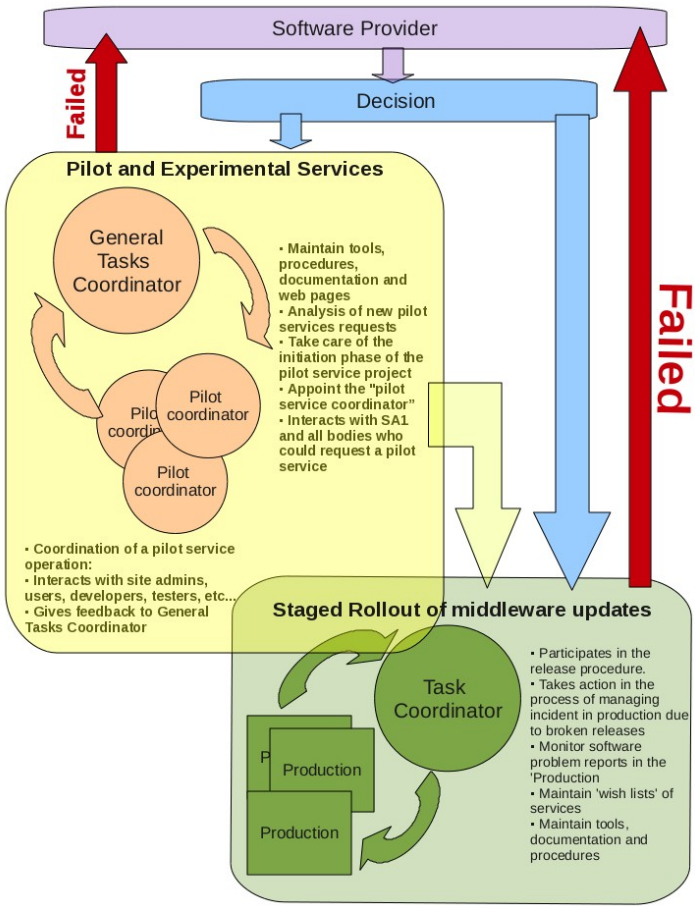

Fig. 3. Detailed Workflow of Middleware through the Staged Rollout or Pilot Services process.

5.2 Operational tools for the middleware Roll-out task

The *Product Teams* will interact/communicate with the *Middleware Unit* using the tools provided by EGI-InSPIRE. Those tools will be under the "*egi.eu*" domain and at least the following ones will be provided:

- *twiki*: documenting all releases with deployment advisories, should have links to more detailed release notes provided by the *Product Teams*.
- *Progress tracking and task tool*: follow all the middleware process from the moment it is declared ready and made available by the *Product Teams* until released into production. The following capabilities should be provided by the tool:
  - Manage all the Staged Rollout process.

    - Creation of task teams for each service in Staged Rollout.
    - Notification capabilities.
    - Feedback to EMI *Product Teams*.
    - Interface with the *Global Grid User Support* (GGUS).
    - Interface with *Grid Operation Centre DataBase* (GOCDB).
- *GGUS*: a support unit for this task will be created. Production sites and Virtual Organization users should use this support unit in case of middleware bugs or issues found in the production infrastructure.
- *Portal*: Including a web form where sites can subscribe as Early Adopters for the Staged Roll-out. It should be either part of the "*Progress tracking and task tool*", or interfaced to it.
- *Repositories*: for all EMI releases. There will be 3 main repositories: *Scratch*, *Beta* and *Production*.

5.3 Operational Level Agreements

In order to guarantee the final Service Level Agreements (SLA) with the Virtual Organizations and Virtual Research Centres, Operational Level Agreements (OLA) should be established following the best-practices defined in ITIL. In this case OLAs should be agreed between:

- EMI Product Teams and EGI Midleware Unit
- EGI Midleware Unit and EGI Operations Unit

The terms of these agreements should be discussed but should cover important aspects like:

- A common Configuration Management Database (CMDB): A database used to store relevant information about each of the middleware components part of Unified Middleware Distribution and the relationships between them.
- Definitive Media Library (DML): One or more locations in which the definitive and approved versions of all the middleware componets are stored.
- Release Unit: Components that will be normally released together.
- Release Identification: A naming convention used to uniquely identify a Release.

5.4 Early Adopters

An Early Adopter is a site which has committed to perform the Staged Roll-out for one or more middleware components or services. The following situations may occur:

- The site deploys a new service in parallel with the production instance, just for the Staged Rollout. process.
- The site may have a clone of the production system, where the new version is deployed, but if some problem occurs, the instance can be quickly changed to the one initially in production.

– The site preforms the Staged Rollout in the production instance itself, having rollback procedures in place in case of problems. S

This decision should be made by the Early Adopter, though it may have input from the Operations Team. Some further consideration should also be taken into account. A node type can be divided into:

1. *Non-Critical*: Computing Element, Workload Management System, clients (User Interface and Worker Nodes).
2. *Critical*: Storage Element and Catalogue services, Virtual Organization Management Service.

Middleware services affecting *Non-Critical* node types are easier to release into a production environment. Problems or bugs that may arise will have in general a limited impact to the users and the site.

On the other hand, middleware services affecting *Critical* node types will have to be dealt more carefully, so has not to lead to data loss or major infrastructure availability blackout.

The middleware services affecting the information system are another component where special care must be taken. At the level of a Top-BDII and the Site-BDII, it may be advisable to have instances running in parallel or cloned from the production one.

Whatever the procedure that the site decides to follow in the Staged Rollout process, it must be taken into account that the site Reliability and Availability should not be affected if problems occur due to the new versions of the middleware components. One way to distinguish services participating in the Staged Rollout, is to tag them as such in the GOCDB.

Most of the sites currently participating Staged Rollout process will form the core of this activity with small or no disruption during the transition phase from EGEE to EGI-InSPIRE. Furthermore, the NGIs that have requested effort in the EGI-InSPIRE for this task, should commit sites to Staged Rollout as early as possible. These two sets of sites should form a stable core of the *Staged Roll-out* process.

There might be sites that will commit to the Staged Roll-out process only for a limited period of time or only for certain releases of some middleware component.

Requesters of new functionalities or new services, if approved, should be engaged in the Staged Roll-out phase eventually committing new sites.

5.5 Other middleware testing processes

The Staged Rollout is the most usual process to release new middleware component versions into the production infrastructure. Nonetheless, other processes for middleware testing may occur:

– Pilot services:

    - Occasionally it may be justified to setup this process when a new middleware component has to be integrated with existing components, introducing the client first and the server only at a later step, with careful testing at all stages. Example: ARGUS the EGEE Authorization Service.
    - Middleware components may need strengthening, improve robustness, test under high loads with configuration tuning. Example: CREAM-CE Computing Element.
  - Testing Alpha and Beta releases of middleware components under development. In particular complex and critical middleware components. Examples: dCache and StoRM Storage Elements, File Transfer Service (FTS).

On any of the cases described previously, there is a tight collaboration between the interested parties; a set of sites, users communities and the developers of the component. It is usual that the versions of these middleware components are deployed and used in production by the participating sites and user communities, before they reach the production repositories for general availability.

Nonetheless, though these components went under heavy testing when they reach the EGI *Scratch* repositories, they should still be put under the Staged Rollout process.

6 Conclusions

The European Grid Initiative is the natural step forward in the implementation of an e-Infrastructure to support the computing needs of the scientific community not only in the Europe but also Worldwide.

The concept of Grid Computing was born more than 10 years ago from the academic world. It has been a long way since the first research and development projects, through deployment of *testbeds* for middleware development, the implementation of the first production infrastructures towards the long term sustainability of these infrastructures.

This paper describes the transition of one of many activities currently underway towards a sustainable and long term support of the Grid infrastructure.

Some of the most important issues are; the transfer of knowledge and the *non-disruption* of the service for the users that rely on the high availability of the infrastructure in their day to day work, as well as for sites (the service providers) in terms of it's operation. This is essential so has not to waste the huge effort and hard work of all who have been involved in these projects over the years.

7 Acknowledgments

This work makes use of results produced by the Enabling Grids for E-sciencE project, a project co-funded by the European Commission (under contract number INFSO-RI-031688) through the Sixth Framework Programme. EGEE brings together 91 partners in 32 countries to provide a seamless Grid infrastructure available to the European research community 24 hours a day. Full information is available at http://www.eu-egee.org.

References

1. "EGI Design Study, EGI Blueprint"; http://www.eu-egi.eu/blueprint.pdf
2. "European Grid Initiative: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe", EGI-InSPIRE proposal
3. http://www.egi.eu/
4. http://glite.org
5. Francesco Giacomini, "Implementing Product Teams in gLite"; https://edms.cern.ch/file/1019948/1/implementing$_p roduct_t eams - v3.pdf$